JANUARY 2004

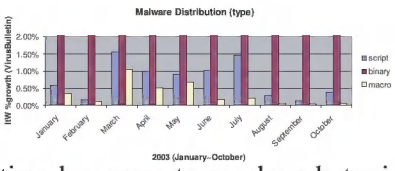
virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

ISSN 0956-9979

IN THIS ISSUE



TRENDS AND FORECASTS

Once again the time has come to ponder what might be in store for the coming year. Jamz Yaneza looks at malware data collected over the past year and attempts some predictions for 2004.

page 14

ALL-ROUNDER

Swen has something for everyone, one might say. Peter Ferrie has the nitty gritty details of this all-action virus.

page 4

HARDWARE SOLUTIONS

Recent reports of a 'new' anti-virus solution based on quick pattern matching of network traffic without a significant reduction in network speed seem tantalising. However, Matthew Wagner has reservations about such a hardware solution.

page 12

vb Spam supplement

In this month's *VB Spam Supplement* ASRG founder Paul Judge explains the aims and objectives of the ASRG, and Pete Sergeant looks at economic and legal solutions to spam.

'Being online is becoming a combination of annoying and worrying experiences.'

John Ogness
H+BEDV Datentechnik GmbH

DRAWING THE LINES

Anti-virus companies have earned the reputation of protecting users against computer-related threats. Today, malicious code has many opportunities to compromise a user's computer. These range from active trickery, such as the use of email, shared files, or instant messengers, to the use of 'invisible' tactics, such as taking advantage of flaws in software.

To complicate the issue, the incidence of spam continues to increase at an alarming rate – currently it is estimated to represent more than 60 per cent of all email. Furthermore, software security holes are uncovered daily and, due to the increasing complexity of new software technologies, they are often more dangerous and more difficult to fix than before.

Being online is becoming a combination of annoying and worrying experiences. So where do the users turn? They turn to the place to which they have always turned for computer protection: the anti-virus community.

Should anti-virus companies extend themselves in order to protect users against these non-virus-related dangers and annoyances? Fighting spam and fixing security problems are nothing new – these industries have been in existence for a decade, though it is only recently that they have been in high demand. From a user's perspective there may not seem to be much difference between anti-virus, anti-spam and general security, but their basic functions have little in common.

Anti-spam is part of a larger industry for regulating information. This includes not only filtering unsolicited emails, but also such tasks as moderating news groups, suppressing pop-ups and controlling web browsing. The focus of this industry is knowing (or learning) what types of information a user wants and regulating incoming information accordingly. Many standalone products are available and a number of operating system and application developers have released products with these capabilities integrated.

The security industry is responsible for collecting security reports, notifying the appropriate individual or organization about any problems, and tracking their resolution. Usually detailed information is made available about the dangers of each problem, how it can be detected, and how it can be fixed. Operating system and application developers often maintain this information for their products and in many cases offer automatic updates to correct the problems. Prevention or handling of computer compromises is another aspect of the security industry.

Anti-virus companies already have the advantage of a reputation for protecting users. But anti-virus companies should not abuse this advantage in order to increase profits or mislead users by releasing their own products with the anti-spam/security label. Just as other industries respect the skills and qualifications of anti-virus organizations, so should the anti-virus industry respect the skills and capabilities of the security and anti-spam organizations that already exist.

This is not to say that anti-virus companies may not participate in these industries. Rather, anti-virus companies should use their role as widely-known defenders to help inform users about the other industries and solutions.

There is a need to build positive relationships between the anti-virus, anti-spam, and security industries. This may require the establishment of standards and interfaces so that their various solutions can work together successfully. It is important that the three industries develop a professional cooperation to help combat the very real and growing threats of today's interconnected society.

Must anti-virus organizations start providing their own security and anti-spam solutions? No, but if they do develop their own solutions, it is important that they acknowledge the fact that they are newcomers to an existing industry, and make efforts to cooperate with rather than re-invent these industries.

NEWS

NUMBER CRUNCHING

Last month saw a flurry of the traditional end-of-year predictions for the security challenges in the year ahead, along with a number of reports estimating the cost of virus attacks to businesses in 2003. However, it is difficult (if not impossible) to truly quantify the 'average' cost of a virus attack with so many variables and subjective issues to be considered. This fact was suitably illustrated by two recent news reports published just one week apart in *Computer Weekly* (*CW* – see http://www.computerweekly.com/). First, *CW* reported that a single virus attack could cost your business £66,000 – a figure arrived at by analyst firm *Datamonitor*. The firm estimated an average cost to businesses of £26,000 for the more 'serious' virus incidents, while 11 per cent of survey respondents said their companies had suffered greater than £66,000 losses from a single incident.

One week later, *CW* reported the results of a different survey, this time carried out by the Corporate IT Forum (Tif). The Tif survey indicated that the 'true cost' per virus attack to UK companies is an average of £122,000 – a figure nearly double that indicated by the *Datamonitor* findings (and more than four times greater than the £30,000 per attack cost estimated by *PricewaterhouseCoopers* and the Department of Trade and Industry's 2002 Information Security Breaches Survey).

Confused? All we can recommend is that figures such as these be taken with a dose of salt. While *VB* recognises the importance of bringing home the notion that virus attacks can and do cause significant losses and upheaval to businesses, it is unlikely that analysts will come up with a reliable formula for putting an accurate figure on those losses.

SAS – THE SYSADMIN SERVICE?

A set of proposals for tackling computer crime has been published by UK Parliamentary lobby group EURIM and the Institute for Public Policy Research. Among other proposals, the paper recommends the introduction of frameworks to facilitate co-operation between industry and law enforcement. One of the ideas put forward is to bring in members of the private sector to assist law enforcement bodies in areas in which they lack the specialist skills necessary to investigate computer crimes. Rather than simply assisting with investigations (as computer security experts have done in the past), the paper proposes that specialists from the private sector would be granted an expanded role and become (unpaid) special constables (without the power to arrest). The full paper can be read at http://www.eurim.org/.

Prevalence Table – November 2003

Virus	Type	Incidents	Reports
Win32/Opaserv	File	7492	27.62%
Win32/Mimail	File	7050	25.99%
Win32/Dumaru	File	2156	7.95%
Win32/Sobig	File	1534	5.65%
Win32/Bugbear	File	1437	5.30%
Win32/Sober	File	1110	4.09%
Win32/Dupator	File	1109	4.09%
Win32/Swen	File	1014	3.74%
Win32/Gibe	File	989	3.65%
Win32/Klez	File	812	2.99%
Win32/Yaha	File	756	2.79%
Win32/Funlove	File	286	1.05%
Win95/Spaces	File	190	0.70%
Win32/Nachi	File	167	0.62%
Win32/Fizzer	File	99	0.36%
Win32/Magistr	File	87	0.32%
Redlof	Script	73	0.27%
Win32/Lovsan	File	73	0.27%
Win32/SirCam	File	59	0.22%
Win32/Holar	File	43	0.16%
Win32/Deborm	File	40	0.15%
Win32/Nimda	File	31	0.11%
Win32/Spybot	File	30	0.11%
Win32/Ganda	File	29	0.11%
Win32/Hybris	File	25	0.09%
Win32/Sdbot	File	25	0.09%
Fortnight	Script	24	0.09%
Win32/Kriz	File	24	0.09%
Win32/Lovgate	File	24	0.09%
Win32/Parite	File	22	0.08%
Win32/BadTrans	File	20	0.07%
Win32/Valla	File	20	0.07%
Others		390	1.03%
Total		27,130	100%

The Prevalence Table includes a total of 280 reports across 74 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

WHO? WHAT? WHERE? SWEN?

Peter Ferrie
Symantec Security Response, USA

W32/Swen spreads in email and newsgroups, across network shares, through peer-to-peer networks, and on IRC. It contains a reasonably authentic-looking message purporting to come from *Microsoft*, and another message that uses an exploit. It hooks file extensions in the registry, and terminates anti-virus and firewall software. Something for everyone, one might say.

INSTALLATION

Whenever W32/Swen is run, it examines its command line. If the command line is empty (which is the case when it is run from an email attachment), the virus will look for its infection marker in the registry.

The infection marker is located in 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer' and consists of a string of pseudo-random upper-case letters, the sum of whose positions in the alphabet (i.e. A=1, B=2, C=3, etc.) is 80. If the infection marker is not found, the virus creates one by constructing a string of random upper-case letters, until the sum of the letters' positions in the alphabet exceeds 53. At this point, the virus adds one more upper-case letter whose position in the alphabet will produce the required sum of 80.

If the infection marker contains a value called 'Installed' whose data are set to '... by Begbie', regardless of the letter case, and if the filename used to run the virus begins with the letter 'p', 'q', 'u', or 'i' – again regardless of case – the virus displays a message which states that 'the update' does not need to be installed on this system.

Otherwise, the virus copies itself to the %windir% directory using a filename of between four and eight random lower-case letters with '.exe' appended. Then it creates the 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run' key in the registry, using a value of four to nine random lower-case letters, whose data are set to the name of the file copied to the %windir% directory. This allows the virus to run whenever *Windows* is started. In order to avoid running this code every time the virus is run, the 'autorun' parameter is added to the registry value data. The virus writes this 'Run' value to a value called 'Install Item' in the infection marker registry key. If the filename used to run the virus

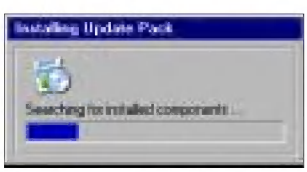
begins with 'p', 'q', 'u', or 'i' (regardless of case), the virus displays a prompt to install a Microsoft Security Update.

The virus will install itself regardless of what the user selects at this prompt, the only difference being that if the user selects 'Yes', the virus will display a dialog box showing its progress.

UNDO THE DEED

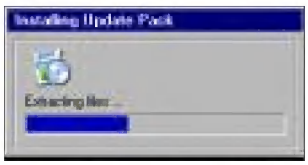
If the name of the computer can be retrieved by the GetComputerNameA() API, the virus will create the files necessary to allow the removal of the virus. First, it creates a file called '%windir%\%computername%.bat', which contains code to run the virus, passing the exact pathname of the virus as the first parameter, and a user-specified parameter as the second.

Next, the virus checks for the existence of a value called 'Unfile' in the infection marker registry key. If this value does not exist, the virus writes to the registry key a string of between four and eight random lower-case letters, followed by '.', followed by three random lower-case letters, and creates the 'Unfile' file in the %windir% directory. The virus updates this 'Unfile' file by appending '%windir%\%computername%.bat'. It is not a bug for the virus to append continually to this file – since it is used for uninstallation, it contains the names of all installed components, thus all files ever created, including those created by partial installations, must be listed.



If the user answered 'Yes' to install the 'update', the virus displays a message, stating that it is searching for installed components.

Now the virus searches for email addresses. The search is conducted recursively in all subdirectories on all hard disks, for files whose extension contains anywhere within it the letters 'ht' or 'asp', and whose filesize is at least 50 bytes; or whose extension is 'mbx', 'dbx', 'wab', or 'eml', and whose filesize exceeds 100 bytes. The virus searches for email addresses within each such file that is found. The contents of files whose extension contains 'ht' and 'asp' are searched for the 'mailto:' string. The other files are searched blindly for text that resembles email addresses. Duplicate email addresses are not added to the list. On completion of the search, and if email addresses have been found, the virus creates a file called '%windir%\germs0.dbv' (if it does not exist already), and appends each address to the file. Once the search has completed, the virus writes 'Yes' to a



value called 'CacheBox Outfit' in the infection marker registry key.

If the user answered 'Yes' to install the 'update', the virus displays a message which indicates that it is extracting files.

EXTRACTING FILES

Despite what the dialog box says, no files are extracted by the virus. In fact, the opposite is the case. The virus copies itself to the %temp% directory, using a filename chosen randomly from one of the following possibilities:

- the letter 'q', followed by either six random numbers or one to four random lower-case letters
- the word 'patch' or 'pack', which may be followed by two to four random numbers
- the word 'update' or 'upgrade', which may be followed by two to four random numbers
- the word 'install', 'installation', or 'installer', which may be followed by two to four random numbers

A 'feature' of the code that produces the strings of random numbers is that it never produces a string that contains '0'. Additionally, the case of the first letter can be altered individually, or the entire string can be converted to upper-case, however the suffix of the filename is always '.exe', regardless of the letter-case of the rest of the string.

After the virus has copied itself to the %temp% directory, it constructs an archive filename of between four and eight random lower-case characters followed by '.zip', and attempts to compress the original file, using *WinRAR* or *WinZip*. The virus attempts to run *WinRAR* first, relying on the *Windows* default search path. If that fails, the virus queries the registry for the 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\WinRar.exe' value. If this value exists, the virus tries to run *WinRAR* from the directory specified in the value data. If this is unsuccessful, the virus attempts to run *WinZip* – first from the *Windows* default search path, then by querying the registry for the 'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\WinZip.exe' value. However a bug exists in this code – the command-line that is passed to *WinZip* is incorrect, and an error will be displayed.

If the archiving is successful, the virus writes the archive filename to a value called 'ZipName' in the infection marker registry key, and updates the 'Unfile' file by appending the archive pathname. The virus then deletes the .exe file from the %temp% directory.

THE DANGERS OF FILE SHARING

If the 'HKCU\Software\Kazaa' registry key exists in the registry, the virus enables file sharing by writing a 0 to the registry value 'HKCU\Software\Kazaa\LocalContent\DisableSharing'. The virus is aware of the different versions of the *KaZaA* file-sharing software, and queries the registry for the 'HKCU\Software\Kazaa\LocalContent\DownloadDir' and 'HKCU\Software\Kazaa\Transfer\DlDir0' values. If either of these exists, the virus will create a number of filenames aimed at enticing people to download them. These are chosen randomly from the following options:

1. One of:

AOL hacker	XP update
Yahoo hacker	XXX Video
Hotmail hacker	Sick Joke
10.000 Serials	XXX Pictures
Jenna Jameson	My naked sister
HardPorn	Hallucinogenic Screensaver
Sex	Cooking with Cannabis
XboX Emulator	Magic Mushrooms
Emulator PS2	Virus Generator

2. One of:

Bugbear	Sircam	Yaha
Sobig	Gibe	Klez

followed by 'remover', 'cleaner', 'removal tool' or 'fixtool'.

3. One of:

Kazaa Lite	Winamp
KaZaA media desktop	Mirc
KaZaA	Download Accelerator
WinRar	GetRight FTP
WinZip	Windows Media Player

followed by 'key generator', 'warez', 'hack', 'upload', 'hacked' or 'installer'.

The filename may be converted to entirely lower-case letters. If archiving was successful, the virus constructs three to seven of these names, appends '.zip', and copies the archived file to these filenames. If archiving was not successful, the virus constructs only one to three of these names, appends '.exe' and copies the original .exe file to these filenames. In either case, the virus appends these filenames to the 'Unfile' file.

The virus also creates a directory in the %temp% directory, using a string of three to seven random lower-case letters, and writes '012345:%temp%\{directory name}' to the 'HKCU\Software\Kazaa\LocalContent\Dir99' registry value. After creating the directory, the virus creates more

.zip or .exe files in this directory, in the same way as described above. The virus completes the *KaZaA* routine by writing 'Yes' to a value called 'Kazaa Infect' in the infection marker registry key.

If the user answered 'Yes' to install the 'update', the virus displays a message which indicates that it is copying files.

COPYING FILES (1)

If 'mirc.ini' exists in the 'c:\mirc' or 'c:\mirc32' directories, Swen creates another filename, using the same routine as for the *KaZaA* routine, copies itself to the %windir% directory using this new filename, and appends the filename to the 'Unfile' file. If 'script.ini' exists in the mIRC directory, the virus moves it to 'script.bcp', then creates a new 'script.ini' file that contains code that will send the file whenever a user joins a channel shared by the infected user.

The virus writes the name of the mIRC directory to a value called 'Mirc Install Folder' in the infection marker registry key. The virus also queries the registry for the 'HKLM\Software\Microsoft\Windows\CurrentVersion\ ProgramFilesDir' value, and looks within the directory listed there for the 'mirc' and 'mirc32' directories. If either of these exists, the virus places a script.ini file in the directory, and updates the 'Mirc Install Folder' value. This is a bug, since only the last mIRC location is saved in the registry, instead of relying on the 'Unfile' file, meaning that some script.ini files remain if the automatic removal is used on machines with multiple copies of mIRC installed.

At this point, the virus queries the registry for the 'HKCU\SOFTWARE\Microsoft\Internet Account Manager\ Default Mail Account' value. Its data are used if they exist, otherwise the virus will default to '00000001'. These data are used to retrieve the SMTP information (SMTP Email Address, SMTP Server, SMTP Display Name) from the registry, by querying the 'HKCU\SOFTWARE\ Microsoft\Internet Account Manager\Accounts\{mail account}' registry key.

If the retrieved email address appears to be valid, the virus writes it to a value called 'Email Address' in the infection marker registry key, and writes the SMTP server name to a value called 'Server' in the infection marker registry key. If the SMTP Display Name cannot be retrieved, the virus will attempt to retrieve the computer name. If the computer name cannot be retrieved, the virus will use the word 'unknown'. The virus will write the computer name (or 'unknown') to a value called 'VicName' in the infection marker registry key.

The virus queries the registry for the 'HKCU\SOFTWARE\ Microsoft\Internet Account Manager\Default News Account' value. If it exists, the virus retrieves the NNTP Server name from the registry by querying the 'HKCU\SOFTWARE\ Microsoft\Internet Account Manager\Accounts\{default news account}' registry key. If the server name appears to be valid, the virus writes it to a file called '%windir%\swen0.dat'.

If the user answered 'Yes' to install the 'update', the virus displays the 'Copying files' message again.

COPYING FILES (2)

The logic in the virus appears to be confused here, since it does not copy files at this time, but does alter the registry, then displays a message referring to registry alterations before resuming file copying.

The virus carries a list of public news servers in a compressed form. The virus extracts the compressed list to a file called '%temp%\nntptmp.fl', decompresses it, writes the decompressed list to a file called '%windir%\swen1.dat', then deletes the compressed file.

The virus author seems to consider the installation to be complete now – the virus writes '... by Begbie' to a value called 'Installed' in the infection marker registry key, then runs another copy of itself with the 'autorun' parameter.

Meanwhile, the virus hooks the 'HKCR\...\shell\open\ command' registry keys for 'exefile', 'comfile', 'piffile', 'batfile', 'scrfile', and 'regfile', and the 'scrfile\shell\config\ command'. The virus is aware of the additional parameter for .scr files. For all but the 'regfile' entry, the effect of the change is that the virus examines the file that was requested to run, and possibly runs it.

For the 'regfile' entry, the virus is run with the 'showerror' option instead. The virus also prevents the use of RegEdit and similar applications in *Windows 2000/XP/2003*, by writing a 1 to the 'HKCU\Software\Microsoft\Windows\ CurrentVersion\Policies\System\DisableRegistryTools' registry value.

If the user answered 'Yes' to install the 'update', the virus displays a message that it is updating the registry. However, the virus replaces this message immediately with a message stating that this update has been installed successfully.

ARE WE THERE YET?

Despite what the message says, the installation has not yet

completed. At this point, the virus queries the 'HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Startup' registry value, then attempts to guess the startup directory on network drives. It tries 'Windows', 'WinMe', 'Win95', 'Win98', and each of the following:

- 'All Users\Start menu\Programs\Startup'
- '\Start menu\Programs\Startup'
- {startup value from the registry} if it is different from above

If a directory is found, the virus creates a filename using a string of five to ten random lower-case letters, with '.exe' appended. The virus replaces the first character with a letter from 'a' to 'v' only (perhaps to reduce suspicion), then copies itself to the directory using the new filename.

The virus also looks on network drives for a directory called 'Documents and Settings' or 'Winnt\Profiles', and tries each of the following possibilities:

- 'All Users\Start menu\Programs\Startup'
- 'All Users\{startup value from the registry}'
- 'Default User\Start menu\Programs\Startup'
- 'Default User\{startup value from the registry}'
- 'Administrator\Start menu\Programs\Startup'
- 'Administrator\{startup value from the registry}'

If a directory is found, the virus creates a filename using a string of five to ten random lower-case letters, with '.exe' appended. It does not replace the first character in this case, before it copies itself to the directory using the new filename.

If the virus is run with the 'showerror' parameter (which occurs when the user attempts to run a '.reg' file to restore the registry), the virus displays a fake error message describing an illegal memory access at a certain location. The numbers are chosen randomly by the virus.

NOT WHAT YOU WERE EXPECTING

If the virus is run with a parameter other than 'autorun', Swen will examine the first and second parameters. If either of these contains anywhere within it any one of a long list of anti-virus and firewall applications, the virus will display the message box as though it were run with the 'showerror' parameter.

The checking of the second parameter (if it exists) allows the virus to detect an application that runs any specified application, since that is one way to bypass malware that changes the 'shell\open\command' registry values. If neither of the parameters contains any of those strings, the virus attempts to retrieve the 'OriginalFilename' field from the version information data in the specified file. This allows the virus to detect renamed files. If the 'OriginalFilename' field contains any of the strings, the virus will also display the 'showerror' message box. It is interesting that 'Gibe' exists in the list: this check works as an inoculation against the W32/Gibe virus ('Begbie' is understood to be the author of both worms).

If the first parameter is the exact pathname of the virus file, and the second parameter is the word 'cure' (regardless of case), the virus will remove itself. This situation can be achieved by running the '%windir%\%computername%.bat' file with the 'cure' parameter.

Removal is performed by terminating any other running copies of the virus, and deleting 'germs0.dbv', 'germs1.dbv', 'swen0.dat', 'swen1.dat' and 'nntpgroups.dat'. The removal code disables file sharing by writing a 1 to the 'HKCU\Software\Kazaa\LocalContent\DisableSharing' registry value, and enables registry tools by writing a 0 to the 'HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableRegistryTools' registry value.

Otherwise, if the first parameter is not recognised, the virus will run the specified file.

A WINDOW ON THE WORLD

The virus uses a window called 'Explorer XBaseBar' to prevent multiple copies of itself running at the same time. If this window does not exist, the virus creates it now and then creates two routines that run periodically.

The first (email) routine is run once every 20 seconds if the SMTP information was able to be retrieved from the registry; otherwise it is run once every two minutes. The virus runs the second (anti-anti-virus) routine now, and every 45 seconds hereafter.

After running the anti-AV routine for the first time, the virus hides itself from the task list on *Windows 9x/Me*, using the RegisterServiceProcess() API, then runs the email routine if the SMTP server name was retrieved successfully. At this point, the virus enters its message loop, waiting for the request to exit.

The anti-AV routine enumerates all processes, looking for process names that contain anywhere within them any strings from the list of anti-virus and firewall names that the virus carries. For each process with a matching name, the virus saves the process ID for later use.

On completion of the enumeration, the virus examines the process ID of each window that exists, and whenever a match is found, the virus sends a request to that window to exit the application. If the request is ignored, the virus terminates the process forcibly.

The virus then checks for the presence of a debugger, using the IsDebuggerPresent() API. If a debugger is found, the virus displays another message.

EMAIL ROUTINE

If no email addresses could be found, the virus will call the routine that searches for email addresses described above. If the search was successful but the SMTP information was not retrieved from the registry, the virus will display a fake error dialog, prompting the user to enter email information.

If the user enters this information, the virus will write it to the registry, then write a string of 11–25 random numbers to a value called 'X-ID' in the infection marker registry key. The 'X-ID' value is added to the emails that the virus sends, which allows the virus to identify its own messages. This ability is exploited by the POP3 routine described below. If the SMTP information exists in the registry, the virus will run the email routine once every 20 seconds.

If email addresses were found, the SMTP information exists in the registry, and an active Internet connection exists, the virus checks whether it has visited a website that (used to) keep track of the (approximate) number of infected machines. If such a visit has not occurred, the virus connects to 'ww2.fce.vutbr.cz', and sends sends 'GET http://ww2.fce.vutbr.cz/bin/counter.gif/link=bacillus&width=6&set=cnt006 HTTP/1.0'. If the server returns without error, the virus writes 'yes' to the 'Counter Visited' value in the infection marker registry key.

If the 'X-ID' value exists in the registry, then once in every 20 times that the email routine is run, the virus connects to the specified POP3 server, examines each message in the mail box, and deletes any message that contains the 'X-ID' text, since it is assumed to be a message sent by the virus.

If the '%windir%\germs0.dbv' file does not exist, the virus will switch to the '%windir%\germs1.dbv' file. If the file can be opened, the virus will read the last 51–80 addresses, the exact number is chosen randomly by the virus. The virus then deletes these from the file. If the file is empty, the virus deletes the file.

Using these addresses, the virus connects to the SMTP server, sends 'HELO' followed by a string of four to eight random lower-case letters, sends 'MAIL FROM' followed by the infected user's email address, then sends a 'RCPT TO' line for each address. Thus, a single mail will be sent, but it will have multiple recipients.

The virus sends the email message at this point, but its content depends on which of the 'germs' files is in use. If the file in use is 'germs0.dbv', the virus will produce the following:

The 'FROM' address may begin with 'MS' or 'Microsoft', and may be followed by 'Corporation'. This is always followed by either:

1. 'Network', 'Internet' or 'Program', followed by the word 'Security', followed by 'Center', 'Section', 'Department' or 'Division'.

or

2. 'Customer', 'Technical', 'Public' or 'Security', followed by 'Support', 'Services', 'Assistance' or 'Bulletin'.

The virus can also choose randomly to replace this part of the address entirely with 'Microsoft' or leave it blank. The address continues with 6–15 random lower-case letters, which can optionally be followed by '-' or '_', followed by four to eight random lower-case letters. Then comes the '@', followed by: 'newsletters', 'advisor', 'support', 'confidence', 'technet', 'bulletin', 'updates' or 'news', which can be followed by '_' or '.', followed by 'msdn', 'ms', 'microsoft' or 'msn'. The address always ends with '.net' or '.com'. The virus can also choose randomly to make this part of the address blank.

The virus can choose randomly to make the 'TO' address begin with none or one of 'Microsoft' or 'MS', which may be followed by 'Corporation' or 'Commercial'. This is always followed by 'Client', 'User', 'Consumer', 'Customer' or 'Partner'. The virus can choose randomly to make this part of the address blank.

The address continues with four to nine random lower-case letters or a word from the list beginning with 'Client' above. This may be followed by '.', '_', or '-', followed by 6–11 random lower-case letters. Then comes the '@', followed by the same text that was produced after the previous '@'. The virus can choose randomly to make this part of the address blank.

The virus can choose randomly to begin the 'SUBJECT' line with none or one of: 'Latest', 'Newest', 'New', 'Current', or 'Last'. This may be followed by one of: 'Internet', 'Network', 'Microsoft' or 'Net'. This can optionally be followed by 'Security' or 'Critical', followed by 'Upgrade', 'Update', 'Pack' or 'Patch'.

The virus may choose randomly to convert the subject to entirely lower-case, or leave it blank. If the 'X-ID' value exists in the registry, the virus will add it to the email at this time.

Next the virus assembles the text for the body of the message. It begins with 'MS' or 'Microsoft', followed by a word from the list beginning with 'Client' above. It continues with

```
this is the latest version of security update, the
[month, year], Cumulative Patch update which
```

and follows this with 'eliminates', 'resolves' or 'fixes', and continues with

```
all known security vulnerabilities affecting MS
Internet Explorer, MS Outlook and MS Outlook Express.
```

This can be optionally followed by

```
as well as three new vulnerabilities
```

or

```
as well as three newly discovered vulnerabilities
```

It is always followed by 'Install now to', followed by one of the following three options:

```
help maintain the security of your computer
help protect your computer
continue keeping your computer secure
```

This is followed by 'from these vulnerabilities', which may be followed by 'the most serious of which could allow an', followed by 'malicious user' or 'attacker', followed by 'to run', followed by 'code' or 'executable', followed by 'on your', followed by 'computer' or 'system'.

This may be followed by

```
This update includes the functionality of all
previously released patches
```

This can optionally be followed by the text:

```
System requirements: Windows 95/98/Me/2000/NT/XP
This update applies to:
      - MS Internet Explorer, version 4.01 and later
      - MS Outlook, version 8.00 and later
      - MS Outlook Express, version 4.01 and later
Recommendation: Customers should install the patch at
the earliest opportunity.
How to install: Run attached file. Choose Yes on
displayed dialog box.
How to use: You don't need to do anything after
installing this item
```

This may be followed by

```
Microsoft Product Support Services and Knowledge Base
articles can be found on the Microsoft Technical
Support web site.
 http://support.microsoft.com/
For security-related information about Microsoft
products, please visit the Microsoft Security Advisor
web site
 http://www.microsoft.com/security/
 Thank you for using Microsoft products.
 Please do not reply to this message.
It was sent from an unmonitored email address and we
are unable to respond to any replies.
 ____________________
The names of the actual companies and products
mentioned herein are the trademarks of their respec-
tive owners.
```

This may be followed by

```
Copyright [year] Microsoft Corporation
```

If the file is 'germs1.dbv', the 'FROM' address will be one of the following possibilities:

1. 'MS' or 'Microsoft', which may be followed by 'Internet', 'Net', 'Network' or 'Inet'. This may be followed by 'Mail', 'Message' or 'Email'. This may be followed by 'Delivery' or 'Storage'. This is always followed by 'System' or 'Service'.
2. 'Postmaster', 'Administrator', or 'Admin'.

The virus can choose randomly to convert the address to entirely lower-case, or make it blank.

After this the 'FROM' address continues with 'post', 'email', 'mail', 'mailer', 'web', 'master', 'smtp' or '[random letter]mail', followed by 'service', 'daemon', 'form', 'program', 'engine', 'routine', 'automat', 'bot', or 'robot'. Then comes the '@', followed by 'rocketmail', 'freemail', 'microsoft', 'netmail', 'bigfoot', 'america', 'aol', 'puremail' or 'yahoo'. The address always ends with '.net' or '.com'. The virus can also choose randomly to make this part of the address blank.

The 'TO' address will begin with 'Network', 'Inet', 'Internet', 'Mail', 'Net', or 'Email', followed by 'User', 'Recipient', 'Receiver', or 'Client'. Once again, the virus can also choose randomly to make this part of the address blank.

The address continues with 'user', 'recipient', 'receiver', or 'client', followed by 'home', 'mail', 'mx', 'smtp', 'your' or 'email', followed by 'domain' or 'server'. The address always ends with '.net' or '.com'. The virus can also choose randomly to make this part of the address blank.

The virus can choose randomly to begin the 'SUBJECT' line with one of the following options:

1. 'Failure', 'Error', 'Abort', or 'Bug', followed by 'Notice', 'Message', 'Report', 'Advice', 'Announcement', or 'Letter'.
2. 'Undelivered', 'Undeliverable' or 'Returned', followed by 'Message' or 'Mail', sometimes followed by ':'. This is always followed by 'User unknown', 'Mailer', or 'Sender'.

The virus can choose randomly to convert the address to entirely lower-case, or make it blank. If the 'X-ID' value exists in the registry, the virus will add it to the email at this time.

Next the virus assembles the text for the body of the message. The text may begin with 'Hi.', and may be followed by

```
This is the qmail program
```

or

```
Message from [from: address, as above]
```

This may be followed by one of

```
I'm afraid
I'm sorry to have to inform you that
I'm sorry
```

This is always followed by one of

```
the message returned below could not be delivered
```

or

```
I wasn't able to deliver your message
```

and followed by one of

```
to one or more destinations.
to the following addresses:
```

This is followed by 'Undelivered' or 'Undeliverable', and may be followed by 'mail' or 'message'. This is followed by

```
to [6-11 random lower-case letters]@[word from list
that begins with "rocketmail"]
```

This may be followed by 'Message follows:'. The virus uses the [MS01-020 exploit], randomly choosing 'x-wav' or 'x-midi' as the MIME Content-Type. The file extension is chosen randomly from: 'com', 'scr', 'bat', 'pif' and 'exe'.

In either case, the virus attaches itself to the email and sends the message now. After sending the message, the virus writes the used addresses to the '%windir%\germs1.dbv' file. There is another bug in the virus here: when the 'germs0.dbv' file is exhausted, the last 51–80 entries in the 'germs1.dbv' file will be used repeatedly.

HERE'S THE BAD NEWS

If no email addresses could be found, the virus will target newsgroups instead. If the '%windir%\nntpgroups.dat' file exists, the virus will read the newsgroup server name from there. If the file does not exist, the virus attempts to read the newsgroup server from the '%windir%\swen0.dat' file, then deletes the file. If the 'swen0.dat' file did not exist, the virus chooses a random newsgroup server from the '%windir%\swen1.dat' file.

The virus connects to the newsgroup server and requests a list of all valid newsgroups, then writes to the '%windir%\nntpgroups.dat' file the newsgroup server name and the names of any newsgroups that contain more than ten messages.

If the newsgroup server allows posting, the virus randomly posts a single message to all newsgroups in the list from the 'nntpgroups.dat' files. The post will appear to come from the 'VicName' value in the registry, if it exists, otherwise it will be the user's name as returned by the GetUserNameA() API, the computer name as returned by the GetComputerNameA() API, or the word 'unknown' if neither of the previous APIs returns a valid value. The name is followed by 6–15 random lower-case letters, which can be optionally followed by '-' or '_', followed by four to eight random lower-case letters. Then comes the '@', followed by between three and five random lower-case letters. The address always ends with '.net' or '.com'. The virus can also choose randomly to make this part of the address blank.

The virus can choose randomly to make the 'NEWSGROUPS' line begin with none or one of 'Microsoft' or 'MS', which can be optionally followed by 'Corporation' or 'Commercial'. This is always followed by 'Client', 'User', 'Consumer', 'Customer', or 'Partner'. The virus can choose randomly to make this line blank.

The 'SUBJECT' line can optionally begin with 'FW:', 'FWD:', or 'RE:', which is always followed by 'Check', 'Take a look at', 'Check out', 'See', 'Prove', 'Watch', 'Taste', 'Use', 'Try', 'Apply', 'Try on', 'Install', or 'Look at'. This may be followed by 'this', 'the', 'these' or 'that', and may be followed by 'correction', 'critical', 'corrective', 'internet', 'security', or 'important'. It is always followed by 'update', 'package', 'patch' or 'pack'.

This can optionally be followed by one of the following options:

1. the word 'for' may be followed by 'MS' or 'Microsoft'. This is always followed by 'Internet Explorer' or 'Windows'.
2. 'that' or 'which', followed by 'comes' or 'came', followed by the word 'from'. This may be followed by the word 'the'. This is always followed by one of: 'M$', 'Microsoft' or 'MS', and may be followed by 'Corp.' or 'Corporation'.

From this point, the rest of the message is exactly as described previously, beginning with 'this is the latest version of security update'.

For each newsgroup in the list, the virus requests all of the news items that exist. If the 'LISTGROUP' command is not supported (it is not defined in RFC 977), the virus requests every news item in a random range between the years 1980 and 1999. The virus extracts the email addresses from each of the news items and adds these addresses to the '%windir%\germs0.dbv' file, which will be used the next time the email routine runs.

CONCLUSION

We were lucky, in a way – the bugs in W32/Swen may have prevented it from becoming as widespread as it could have been. Now that the *Microsoft* patch email trick has been played, people won't be fooled by that anymore. Or will they? Some people do have short memories …

LETTERS

JOURNALISTIC INTEGRITY?

The article 'Microsoft, monopolies and migraines: the role of monoculture' by Richard Ford in the December 2003 issue of *Virus Bulletin* (see *VB*, December 2003, p.9) raises a serious editorial issue. The web page http://www.se.fit.edu/partners/index.html clearly shows that Florida Institute of Technology's (FIT) research activity is sponsored by *Microsoft*.

Virus Bulletin failed to mention or signal this fact in print, which casts a shadow on the quality of journalism that goes into your publication. Please would you publicly clarify this, because it is significant towards balanced evaluation of the *preliminary* research results disclosed by Mr Ford.

Tamas Feher, 2F 2000, Hungary

IN RESPONSE

[*VB asked Richard Ford to respond to Mr Feher's concerns, his response follows - Ed.*]

As a leading centre worldwide for security research, the Center for Information Assurance at Florida Institute of Technology has a long and distinguished list of sponsors. However, for the record, *none* of my research to date has been funded by *Microsoft*.

Notwithstanding, even if the research had been funded by *Microsoft* this fact would be far less important than the article's scientific validity. The issue of monoculture *must* be debated objectively, not subjectively. Only by examining the facts in an objective manner can we hope to come to a conclusion that is untainted by feeling. I would welcome someone questioning my conclusions; questioning of motive is likely to generate heat, not light.

Dr. Richard Ford, FIT, USA

CALL FOR PAPERS

VB2004: CALL FOR PAPERS

Virus Bulletin is seeking submissions from those wishing to present at VB2004, the Fourteenth Virus Bulletin International Conference, which will take place on 30 September and 1 October 2004 at the Fairmont Chicago, Illinois, USA.

While past *VB* conferences have been focused exclusively on anti-virus technologies and malware threats, VB2004 will also cover spam and anti-spam techniques. Submissions are invited on all subjects relevant to the anti-virus and anti-spam arenas. The following is a list of suggested topics elicited from attendees at VB2003. Please be aware that this list is not exhaustive and papers on these and any other AV and spam-related subjects will be considered.

- Hardware AV solutions.
- Detailed discussion of the latest viruses.
- Control of web-based transmission of malware.
- P2P threats.
- Vulnerabilities and patch management.
- AV engine architecture.
- Hoaxes and spam from a legal point of view.
- International computer crime laws.
- How AV applies to or fits in with Critical Infrastructure issues.
- Cybercrime, malware intelligence gathering and legal issues associated with catching virus writers.
- Forensics: tools, techniques, reading IP headers etc.
- Virus/worm traps on internal networks.
- Threats relating to the *.NET* framework, *IIS6.0,* XML.
- *Linux* security issues.
- AV within *MS Exchange 2003*.
- Corporate case studies of single virus incidents.
- Corporate case studies of spam management.
- Implementing a successful corporate anti-virus strategy.
- Integrating anti-virus, anti-spam, IDS and other security software.
- Prevention of fast-spreading, 'Slammer-like' malware.
- Trends in the evolution of viruses.
- Use of VMWare for malcode testing.
- Security issues relating to PDAs and mobile phones.
- Central management of anti-virus (e.g. ePO) and the lessons learned.
- Ethics – what makes for a good code of ethics for users?
- Corporate end-user training. Corporate virus response team training.
- Spyware, RATS, adware, hacker tools, DoS tools.

Abstracts of approximately 200 words must reach the Editor of *Virus Bulletin* no later than **Wednesday 31 March 2004**. Submissions received after this date will not be considered. Abstracts should be sent as RTF or plain text files to editor@virusbtn.com. Further details of the paper submission and selection process are available at http://www.virusbtn.com/conference/ along with more information about the conference.

FEATURE

HARDWARE ANTI-VIRUS SOLUTIONS?

Matthew Wagner
Florida Institute of Technology, USA

Scanning the computer security newsfeeds, one periodically runs into stories that seem tantalising. Most recently, the press has reported a 'new' anti-virus solution based upon quick pattern matching of network traffic without a significant reduction in network traffic. This work comes not out of industry, but from a team led by John Lockwood of Washington University.

Described in detail in the paper 'Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware' [1], the system consists of Data Enabling Devices (DED), a Content Matching Server (CMS), and a Regional Transaction Processor (RTP). The data enabling devices are placed at network aggregation points and use field programmable gate arrays (FPGA) to scan the traffic content. New search strings are added to the system through the content matching server, which reprograms the DEDs dynamically. Finally, if matching occurs on a particular search string the regional transaction processor is consulted as to whether it should block or forward the traffic.

The benefit of the hardware approach over software-oriented scanners is that, due to the parallelization capabilities of hardware, multiple content matches can occur simultaneously [1]. Software scanning and matching, Lockwood contends, significantly affects network throughput as the volume of traffic increases.

The technology is remarkable and can certainly be applied to a number of areas where pattern matching on network traffic is needed. Recently, however, the system has gained attention as a potential solution to the problem of malicious code [2] [3]. Used in this respect, the DEDs would scan network traffic for the signatures of malicious code and, depending upon the system configuration, block the traffic from entering the network. To those not familiar with malicious code, the system sounds as if it would be effective. In reality, however, it does not solve the general problem, and in some ways is at a disadvantage compared with existing detection methods.

EPIDEMIOLOGY, COUNTERMEASURES AND PROPHYLAXIS

In order to gauge the applicability of any system with regard to detecting and blocking malicious code, it is important to understand the big picture. Current anti-virus software (when maintained correctly) is highly effective in stopping known threats, and can usually contain an outbreak once it has occurred.

Despite the advent of heuristics and other methods beyond signature scanning, the actual process for responding to an outbreak has largely remained the same. The virus, worm, or Trojan is discovered in the wild and analysed by anti-virus vendors and researchers. Next, either the anti-virus vendors develop a detection method which can be employed by their product or the software company responsible for the vulnerability develops a patch. Finally, the update is uploaded to servers from which customers may download it and apply it to their systems.

The process has proved fairly effective at stopping an outbreak *after* it has occurred, although it fails to stop the outbreak from occurring in the first place. Therefore, the unsolved problem within the anti-virus research community is not cleaning up after an outbreak, but rather preventing the outbreak from occurring in the first place, or slowing it to such a point that reactionary measures can be deployed before an infection becomes widespread.

CHALLENGES

Unfortunately, while the system may well offer significant benefits in terms of throughput and scalability, it does not seem to possess proactive capabilities. The process described above is only slightly different for the proposed solution in that, instead of downloading a patch to the client machines, the signature is added to the system manually by someone with appropriate permissions.

Clearly the proposed solution is at a disadvantage because it requires user interaction. It takes minutes to update, during which time the system is inoperative. In addition to failing to address the general problem, it can be argued that the additional user interaction required may, in many instances, amplify the problem. This task would most likely be the responsibility of an IT department and it simply is not realistic to rely on the 'typical' IT department (often running months behind on applying critical updates and patches to their system) to update their signature scanning devices quickly enough to prevent the malicious code from entering their network.

Should the system be automated, such that anti-virus companies can patch the devices of their customers, the user interaction delay would be eliminated – although the time taken for the entire process would still be comparable to that of traditional methods. However, vendor updates may not work with local site-specific updates and it is questionable as to whether organizations would allow a third party to manage such a device on their network. Privacy and trust issues arise, as these devices have the potential to scan and

block legitimate traffic as well as viruses. Network administrators might not feel comfortable with the idea of a third party scanning their traffic, since confidential and proprietary information may be passed through the system.

Furthermore, improper configuration of the device by a third party would have the potential to cause serious network outages. Whereas current software-based anti-virus solutions are also susceptible to errors in the patching process, the effect remains localized to the systems on which the product is installed, rather than affecting the network as a whole.

Despite the system failing to address the general problem of malicious code, it may still be effective at detecting known threats and containing outbreaks after they have occurred. Unfortunately, the system also fails to account for issues which make the detection of known threats more difficult at network aggregation points as opposed to endpoints.

One of the problems regarding the detection of known threats is that the DEDs implement signature scanning as their sole detection method. Although this method can detect simple malware accurately, authors of malicious code have for years overcome this primitive defence through the use of encryption and polymorphism.

Signature scanning relies on the fact that the virus will make an exact copy of itself every time it attempts to spread. Initially, malware authors responded to signature scanning through the use of encryption, in which each copy of the code was encrypted with a unique encryption key. Although the body of the code appeared differently in each copy, the decryption routine remained constant. As a result of this weakness, polymorphic viruses were developed in which the decryption routine itself appeared differently in each copy of the virus.

The combination of encryption and polymorphism resulted in viruses from which no reliable signature could be derived. Software-based anti-virus solutions deal with polymorphic viruses by loading the virus into a virtual machine, allowing the virus to decrypt itself, and proceeding to apply signature scanning, heuristics, and characteristic checking for items unique to each polymorphic virus. Polymorphism is no longer limited to viruses and polymorphic shell code has been produced in order for exploits to circumvent IDS systems.

In addition to the issues arising from polymorphic code, it is assumed that the traffic passing through the DEDs will be unencrypted, uncompressed, or not encoded in any way. In relation to signature detection, this is a critical oversight because even if the malware does not implement polymorphism to hide its signature, it may still pass through the system as the result of line encryption, compression or encoding.

Line encryption is increasingly gaining popularity as a defence against network sniffers. For instance, Telnet has given way to SSH and many mail servers now implement SSL for SMTP encryption. Should the traffic passing through the DEDs be encrypted through IPSec, SSL, or any other line encryption method the signature of the malware will be unrecognizable.

Due to the fact that the raw traffic is scanned without performing any sort of decryption, malicious code contained in encrypted traffic has the potential to flow easily in and out of the network undetected. Similar to the problem of encryption, if the malware is compressed or MIME-encoded when it is sent across the wire the signature may also be unrecognizable. Current anti-virus solutions, which operate at the network endpoints, have the advantage of the traffic having been decrypted, and have the ability to uncompress and decode MIME-encoded attachments prior to attempting detection without significantly affecting the throughput of the entire network.

Aside from the detection issues, blocking malware using this method may also be undesirable. While the means by which blocking occurs are not described in Lockwood's paper, it can be assumed that it occurs at a low level – possibly at the data link layer. Even though this effectively prevents detected malware from reaching the client, it can have unforeseen consequences such as dropped sessions and retransmissions. Software solutions on the endpoint, however, are able to act at the presentation layer, at a much reduced risk to the correct operation of the network.

CONCLUSION

In summary, the failure to address the general problem, polymorphic code, and the inability to deal with encrypted, compressed, or encoded data are all prime reasons why the system in its current state is far from a cure-all for preventing future malware outbreaks and may be ineffective in containing outbreaks. Each of the problems can be attributed to the fact that the system does not process the data aside from simply scanning for a sequence of bytes or a particular regular expression.

Anti-virus software has moved far beyond simple signature scanning in order to detect polymorphic code and to take a more proactive approach to combat the issue of reaction time. Due to the rate at which malware can spread, purely reactionary systems – which are only effective after the outbreak has occurred – will not be able to prevent future outbreaks. In addition, solutions which are deployed at the network aggregation points are at a distinct disadvantage in this regard because further analysis of the data, beyond signature scanning, could cause a considerable slowdown in network throughput and performance.

In contrast, traditional solutions deployed at the endpoint have the advantage of being able to process and analyse the data using a more thorough and complete method. Until detection at network aggregation points evolves such that a proactive approach is feasible without a significant reduction in overall network performance, endpoint detection will remain more effective.

While the ability to perform pattern matching on network traffic without a significant reduction in network speed or throughput is exciting and the system as a whole is innovative, it will not help prevent future outbreaks of malicious code and, depending upon the nature of the threat, may not be effective in containing future outbreaks. In its current state, Internet worms, viruses, and Trojans will still have the potential to infect networks that are 'protected' by the system. Until the system is expanded so that it boasts the capability to detect polymorphic code and the ability to decrypt, uncompress, or decode data, it will remain less capable in containment of malware than endpoint-oriented anti-virus solutions. Simulation of potential impact of the solution in its current form would be an interesting excercise: given the epidemiology of typical mass mailers, for example, what suppression levels could be gained over server-based detection?

In regards to the general problem of outbreak prevention, if the system becomes automated and administrators are not fearful of the privacy and trust issues it may eventually be comparable to current solutions.

Although the system may be inadequate for detecting and blocking malicious code, additional uses such as copyright protection, trade secret protection, and transaction documentation have been discussed elsewhere, and may prove better suited to such technology [1]. Each of these areas should be explored further, though the challenge of encryption still rears its ugly head.

In conclusion, until a more robust solution for detecting and blocking malicious code at the perimeter of the network is developed which looks beyond primitive signature scanning, the current methods employed at network endpoints will remain more effective.

References

[1] John Lockwood, James Moscola, Matthew Kulig, David Reddick, Tim Brooks, 'Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware', Applied Research Laboratory, Washington University, 2003.

[2] Tony Fitzpatrick, 'New System Halts Malware', *E4Engineering.com*, November 2003. See http://www.e4engineering.com/item.asp?nid=f4sw8&id=50510.

[3] Rupert Goodwins, *Alternative Medicine: Future Virus Fighting*, zdnet.com November 2003. See http://insight.zdnet.co.uk/specials/viruses/0,39025060,39118047,00.htm.

OPINION

THE MALWARE BATTLE: REFLECTIONS AND FORECASTS

Jaime Lyndon 'Jamz' A. Yaneza
Trend Micro, Philippines

Another year has come to its end and the malware battle still rages on. It seems to be a never-ending uphill struggle to secure digital information.

By now most enterprises will at least have some form of sentinel guarding their interests, but is it enough? Even as content management solutions that include improved anti-virus, firewall, or other security innovations are developed, the malware landscape continues to evolve. With corporate spending budgets the focus of attention, the question is: how do system administrators forecast their defensive position and provide data to upper management?

Data is usually subjective in terms of the geographic location and period of time over which the information is gathered. Statistical data for a given period will not indicate the development direction that virus writers are taking. Forecasts or predictions should also be based on the outbreaks seen worldwide, along with analysis of the specific details of each outbreak.

Looking at the raw data collected by *Trend Micro* for the busiest months in a three-year period from 2001 to 2003, it can be seen that the number and type of outbreaks observed from 2001 through to 2003 are relatively similar.

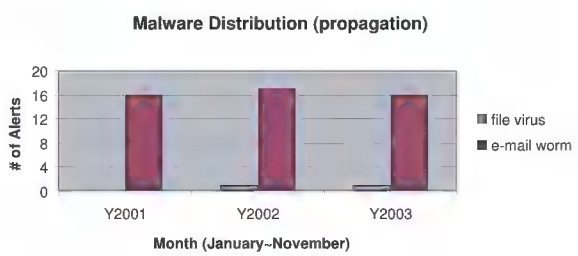


Figure 1. Source: http://wtc.trendmicro.com/wtc/.

Mass-mailing worms are here to stay as the current malware of choice. The standard use of mass-mailing capabilities is the effect of a more inter-connected digital world as well as virus writers having discovered a way to propagate their malicious creations further and faster by wholly depending upon users' bandwidth.

Further scrutiny of the data shows that outbreaks caused by script and macro viruses dropped lower into the charts at the onset of 2002 and had virtually disappeared by 2003. A similar snapshot of data from the *Virus Bulletin* virus prevalence tables over the same time period shows the

percentage of the different basic types of malware in the Wild (ItW).

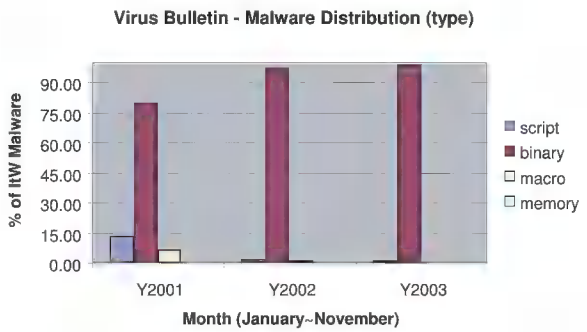


Figure 2. Source: http://www.virusbtn.com/prevalence/.

Observations from *Trend Micro*'s month-to-month comparison of malware-type distribution for the year 2003 show that, on average, scripts, binary executables, and macro viruses account for 16%, 70%, and 14% of malware respectively. It appears that infection growth levels of the basic malware types have stayed more or less the same during 2003. Over approximately the same month-to-month period, the *Virus Bulletin* prevalence data shows a more pronounced differentiation, but more or less matches the rise and fall pattern.

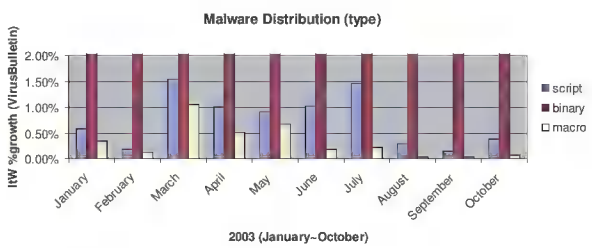


Figure 3. Malware type distribution in 2003.

More information can be gleaned by sifting through the malware types of script, binary, and macro data separately. It is notable that batch file and mIRC script statistics almost match one-for-one owing to malware that attempted to stay resident on the system by cross-dropping its installations. The number and distribution of macro viruses in the Wild reflects approximately the everyday usage of the relevant platforms (see Figure 4).

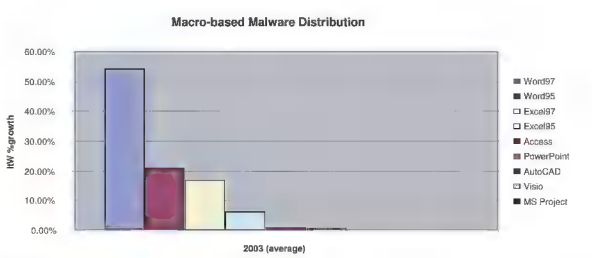


Figure 4. Macro-based malware distribution (statistics for different software versions have been merged, except for MS Word and Excel).

Although only showing up as blips on the radar for now, reports of adware/spyware and Macintosh malware are evident in 2003 (see Figure 5). For those still foolish enough to believe that malware does not exist on *Linux* it is interesting to see the script values added to binary numbers. Trojan-based malware programs that optionally install backdoors are seen in the greatest numbers.

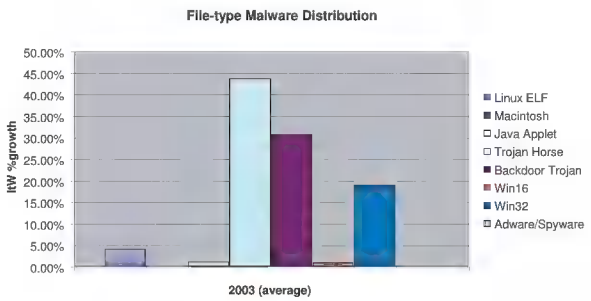


Figure 5. File-type malware distribution.

The use of Internet relay chat (IRC) emerged as a vector of malware distribution in 2002 and a blip or two in the 2003 radar. An interesting individual case we encountered was a large corporate-wide infestation of a backdoor Trojan installation which baffled administrators as it did not have any worming capabilities – only later did they discover that several employees had been connecting to a rogue chat server installation which was accessible externally.

Exploits that abuse system vulnerabilities such as those on *Microsoft Internet Information Service* (IIS) and Apache, proof-of-concept malware on *Microsoft SQL Server*, and various exploits causing auto-execution of email attachments appear to be rising interests as well.

Although the use of mass-mailing features shows a decline due to better attachment filtering practices, it is still the most effective distribution method when coupled with a little social engineering. Mapped and system shared drives are even now becoming a propagation standard – probably

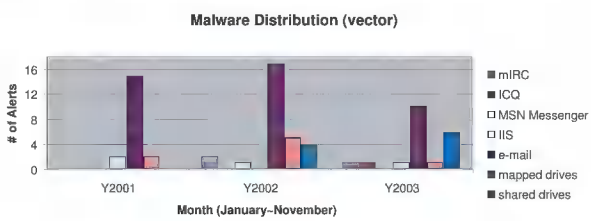


Figure 6. Malware distribution vectors.

due to lapses in proper configuration or security with a notable Share Level Password vulnerability affecting *Windows 9x*-based installations. The term *blended threat* has been coined to refer to these types of malware that combine several attack vectors (see Figure 6). When forecasting protection strategies based on the chart above, administrators should be well aware of the unique characteristics that malware programs adapt to ensure their own survival in a penetrated corporate environment.

RECAP

As a summary, our recap of 2003 includes the following observations:

1. Mass-mailing worms are using email with some form of social engineering to entice users to click and execute attachments.
2. Self-compression and encryption coupled with anti-debugging code is a growing concern as it adds another layer of complexity, making it harder to analyse a piece of malware.
3. Vulnerabilities and bugs in commonly used software are proving to be the Achilles' heel of protection strategies and as such are becoming favourite tools in hackers' and virus writers' arsenals.
4. There was a noticeable increase in malware employing Denial of Service attacks in 2003, a resurge from 2000.
5. Depending on what elevated user privileges a compromised system provides, backdoors may allow hackers to cause prolonged damage.
6. The use of self-installing malware URLs to pull down updates and components from hacker-compromised Internet locations has proven to be an emerging technique. A simple link combined with ActiveX code can pass through anti-virus and filtering software to be clicked on by the unsuspecting user.
7. Another common characteristic of current malware is the use of self-checks to ensure parasitic presence as well as to disable and unload the running anti-virus, personal firewall, and anti-Trojan monitoring software running in system memory.
8. There is a trend towards packaging malware in archives in order to avoid attachment filtering at the email gateway.
9. Virus writers are now packaging their creations with their own SMTP engines, thus effectively eliminating the dependency on the MAPI used by *Microsoft*'s email solutions.
10. It seems virus writers also learn from their mistakes and are going back to pure virus basics – for example by doing away with destructive payloads.

WHAT'S NEXT?

The bottom line is: what's next? Based on all the facts observed and those presented here, it would be safe to make the following predictions for 2004:

- The use of 'blended threats' to attack networks will remain the present standard.
- Current and future malware will continue to attempt to disable anti-virus, personal firewall, or even anti-Trojan monitoring programs.
- Web-filtering software, or at least Internet surfing policies must be put into effect in corporate environments to prevent inadvertent redirection to malware-related websites.
- Email attachment filtering will continue to provide add-on protection. However, gateway scanning anti-virus software is more efficient at weeding out infected files passing through corporate networks as well as recognizing different types of archive and file format.
- Common public and unmoderated messaging channels such as IRC and P2P will be used increasingly given the increasing need for faster communication as the email glut continues to pound day-to-day operations. Proper port configuration needs to be considered.
- Anti-spam legislation is a hot topic and enterprises should be prepared.
- As enterprises grow the use of centrally managed services becomes more important. Several vendors offer content management solution packages and these may deserve more than a cursory look. Administrators must be careful to note their overall efficiency and ability to provide collaborative data.
- Management tools with the ability to isolate malware-infested segments of a corporate network and the ability to retreat to a safe ground of core functionality will be important capabilities to look for.
- Continuous user education is a must. Corporations will also need to look to provide policy enforcement to ensure secure environments.
- System administrators must be careful in evaluating and considering the general software needs of their corporate network. Criteria should include software whose developers can at least commit to fixes to vulnerabilities on time as well as services that can be delivered reliably and consistently.

PRODUCT REVIEW

AUTHENTIUM COMMAND ANTIVIRUS FOR WINDOWS ENTERPRISE 4.90.2

Matt Ham

This product bears the name of *Authentium*, is derived from the *FRISK F-Prot* engine and historically has been produced by *Command Software* and known as *Command AntiVirus*.

The *Authentium* brand covers a wide range of products, although the general theme is security with some patch management applications thrown in for good measure. The *Command* product line has long been part of the line-up in *Virus Bulletin* comparative reviews, though its link with *Authentium* is more recent. In addition to the desktop and server products which are standard offerings amongst anti-virus product ranges, there is also the ability to integrate at the ISP level.

Included in the material submitted for testing was the SDK (software development kit) for *Command*'s anti-virus functionality. Over the last year or so, the number of manufacturers offering this variety of product has been increasing steadily. Many developers are heading towards (or have reached) a more modular construction for their products – by having a platform-portable detection engine, the development issues for a new platform are confined primarily to supplying a new front end. A side-effect is that it is easy to offer the same detection engine to third parties, who can construct their own interfaces for the APIs supplied.

WEB AND DOCUMENTATION

The pure anti-virus side of the *Command* offerings is somewhat overshadowed on the *Authentium* website (http://www.authentium.com/) by the large number of other security products on offer. These are classified by means of various domains: .edu, .mil and the like, with anti-virus fitting into the .biz category for reasons which appear pragmatic rather than strictly taxonomic. The website is rather jumbled as a result of this division of products by potential customer rather than by the function of the products.

Documentation is available either on physical media or from the website in the same format. For most products this consists of a text format quick-start guide and a more detailed manual in PDF format. During the review the functionality of the product was found to be sufficiently intuitive to operate without reference to outside material. However, the manual does contain a large amount of information which explains more complex issues such as network deployment and administration. The documentation is thus rather more useful for administrators than for end-users.

INSTALLATION

The version tested was supplied as a self-extracting executable which was 16 MB in size and which, upon execution, produced an MSI file and associated objects.

Execution of the MSI file starts the installation process, with a readme screen being the first item of note. This is probably the longest of its kind that I have seen. The readme includes information on the platforms supported, installation methods, documentation, version history for the last six months or so and contact details. It is difficult to determine what is actually important out of the vast amount of information provided. The readme screen is followed by a rather shorter licence agreement which must be accepted before the installation proper can commence.

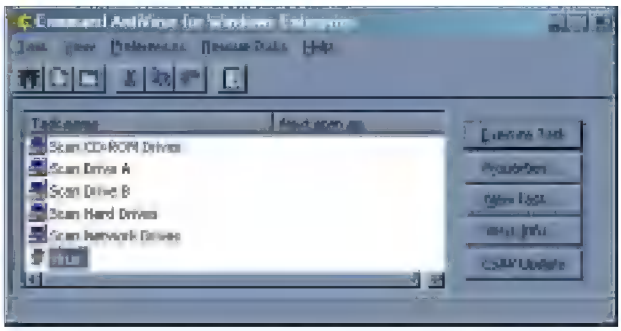
Next comes the choice of update mode – updates may either be obtained directly from the Internet or supplied by a network server. Due to the deliberately isolated nature of the test network, the option to update via the network was selected. The next choice is as to whether Typical or Custom features are installed. If the Typical option is selected, the installation proceeds to completion, without displaying details as to exactly what has been installed.

The Custom feature selection offers a much better idea as to what the installation will include. There are three categories: Command AntiVirus Scanner, Dynamic Virus Protection and Optional Files, these being for the on-demand, on-access and non-scanner portions of the product respectively.

The Optional Files are further subdivided, with NetWare Reporting, Internet Update, Outlook Scanner, Scheduled Scan and Shell Extension being the options.

In the Typical installation the Command AntiVirus Scanner, Dynamic Virus Protection, Scheduled Scan and Shell Extension features are selected. In the case of the Optional Files, the files themselves may in fact be positioned on remote machines for use across a network. When adjustments have been made to the required features, installation can be triggered. Rebooting was not required under any of the configurations tested.

Although theoretically the installation is complete at this point, in practice patches were required in order to upgrade the product fully. The latest patch available was 2 MB in size and fixed several problems (one of which had emerged in preliminary testing). In a similar manner to the base



product, the patch was supplied as a self-extracting file which produced an MSP file – MSP files are patches for MSI-installed applications. What is not mentioned in this case is that the original MSI file must be available so that patching may complete. If the MSI file is not found an error is thrown up part way into the patch process, and patching is aborted.

In the readme file packaged with the MSP file it is noted that a signature update should be performed after patching has been completed. The product was thus updated with one of the executable updaters available from the *Authentium* website.

When reviewing the *FRISK* product last month (see *VB*, December 2003, p.14), which is based on the same *F-Prot* engine, a note was made concerning the definition files. For *FRISK* products these are split into two portions, each of which will fit onto a 3.5-inch floppy disk. In the case of *Authentium*'s updates, however, the executable update file is a monolithic 1.6 MB in size.

As might have been predicted, launching this executable extracts a further MSP file, which is activated for the update process. Once again, the process requires the presence of the original product MSI file. It was discovered that if this file is not available it is possible to deactivate control of on-access scanning by attempting to use the patch. However, this slight irritation was easily solved by a reboot.

FEATURES

The standard product submitted for testing was the desktop scanner, tested on *Windows 2003 Server*. This manifests itself in only three ways once installed: the on-demand scanner available through the start menu, the on-access controls on the task bar and the AntiVirus Scan option available through the right-click menu.

The interface provided by the on-demand scanner is simple and uncluttered, though rather irritating on one level, in that it cannot be resized. The initial view offers a variety of preset tasks, these being triggered through double clicking, the use of right-hand mouse buttons, the use of icon buttons or through drop down menus. In this initial view the controls for viewing, copying, editing and creating new customised tasks are available. Additionally, updates and virus information applications may be triggered. The term 'virus information' is a little misleading here, since all that is available is a list of the viruses detected by the current version.

In addition to this initial view the scan results may be viewed. With a small number of alternative views and few visible controls it is the menus in the initial view which allow customisation of the product.

The drop down menus cover the headings of Task, View, Preferences, Rescue Disks and Help. Task options have already been mentioned, while the view menu simply allows switching between the two views.

The Preferences menu offers many more options. Starting with Reporting, there are options to sound beeps when viruses are detected, list all scanned files and report to the application log. These options are all disabled by default. SMTP mail alerting is also available to single or multiple recipients, and messages upon detection may be customised. However, it is not possible to change to the location to which logging occurs.

The Dynamic Virus Protection entry controls on-access scanning. By default this is set to disinfect any infected files that are found and remove all macros if a macro virus is found, but not absolutely identified. By default all areas are selected for scanning – though, if required, scanning may be disabled for floppies, hard drives or network drives. The usual range of actions is available if infections are detected.

The choices available under the 'Files to Include/Exclude' menu entry are similarly predictable, with the options being to exclude either files or directories from on-demand scanning. Although the default option in the scanner is to scan all files, *Command AntiVirus* can also use a list of extensions to determine what will or will not be scanned. This includes the enigmatic entry of '{??', which is presumably the entry for blank-extensioned files, and extensions may be added or removed from the list as required.

The Advanced menu entry is the next for perusal. In a fresh installation this is rather difficult to use. The initial tab under this category is that where the quarantine directory is selected. The settings for this point the quarantine directory to an as yet uncreated directory which causes an error message to be triggered if any attempt is made to leave the tab – whether by selecting another tab in the same dialog or through closing the dialog. This is easily fixed by creating the directory or by selecting another which does exist, but is

irritating nonetheless. The other tabs available here offer warnings for out-of-date signature files and the option to select the storage location of the scanning task configurations.

The last entry in the Preferences menu is for the configuration of updates. This offers almost as many settings as all the other Preferences menu items combined. Updating may be scheduled for certain times, with control over what happens if an update fails, whether updates will occur only during idle time and a variety of options relating to power management, which will be of use to laptop users.

The remaining drop down menus cover the creation of rescue disks and a selection of informational options under the Help menu. The Help function is inoperational on a standalone machine, since it opens the documentation on *Command*'s website.

SCANNING

Creating a scanning task is simply a matter of choosing a name for the task and deciding whether the task will run as a system task or be available only to the particular user. After these selections a dialog box appears, where action on detection and scan targets are selected. The choice is also available to use the extension list rather than scanning all files, and if this is selected then scanning of compressed or packed files may be disabled. By default, files which have been quarantined are not scanned – this setting may also be changed here.

With a distinct dearth of scanning options to play with, only a few scanning tests were performed. Initial tests did produce some excitement, however, since it appeared that, when on-demand scanning was initiated, the files being scanned were also scanned by the on-access scanner. This led to infected files being blocked by the on-access scanner, labelled as clean by the on-demand scanner, yet remaining infected. Fortunately, however, this bug was addressed by a subsequent patch and does not occur in the current fully-patched version of the product.

The on-access scanner's default mode specifies that disinfection is the primary choice of action upon discovery of infected files, but not what occurs when worms or non-disinfectable executables, for example, are detected. Experimentation determined that these files are deleted and denied access respectively.

As far as detection was concerned, three samples in the polymorphic test set and a single sample of JS/Unicle were missed in the entire test set. This was much as was expected as *FRISK*'s *F-Prot* had missed the same polymorphics in previous tests. It is unlikely that these misses will recur, however, since two are detected by the most recent version of the *F-Prot* engine.

THE SDK

The SDK was provided as a central application together with a collection of scripts demonstrating key functionality. The application is simply executed and thereafter provides APIs which may be accessed to provide anti-virus functionality. The SDK is a quarter of the full product's size at 4 MB. Unfortunately full documentation was not available for the SDK and thus the existing example scripts were examined to give an idea as to how easily these APIs could be accessed. Of particular interest was a 14 KB script offering a fully functioning, if rather basic, on-demand scanner.

Of this script the bulk is concerned with setting up variables for the scan parameters, these all having fairly self-explanatory names. It became apparent that there are also settings available from within the APIs that are not available when using the standard GUI version – for example the setting 'UseArtificialIntelligence' seems to have no parallel in the commercial on-demand scanner. Although detailed testing of the SDK's functionality was not performed, the interface to the engine seemed relatively simple and there seemed to be no obvious gaps in its expected functionality.

CONCLUSION

Authentium's product is most notable for its lack of possibly extraneous options. Limiting the scanner to only one type of scan, without the ability to alter heuristic settings or other such matters, is one with both pros and cons. For an average user the lack of confusion will probably be welcome, though more advanced or inquisitive users may find this setup to be slightly limiting. Since *Command AntiVirus* is built using the *F-Prot* engine, which supports such options, and the options are visible in the SDK, it is clear that *Authentium* believe that the simplification of the scanning process is worth the slight decrease in functionality.

Technical details:

Product: *Authentium Command AntiVirus for Windows Enterprise 4.90.2.*

Test environment: Identical 1.6 GHz Intel Pentium machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-Rom and 3.5-inch floppy drive running *Windows Server 2003 Web Edition V5.2 Build 3790.*

Developer: *Authentium, Inc.*,1061 East Indiatown Road, Suite 500, Jupiter, Florida 33477, USA. Tel +1 561 575 3200; fax +1 561 575 3026; email sales@authentium.com; website http://www.authentium.com/.

END NOTES & NEWS

Black Hat Windows 2004 Training and Briefings take place in Seattle, WA, USA 27–30 January 2004. Meanwhile, the call for papers for the Black Hat Europe Briefings (Amsterdam, Spring 2004) is now open, and a call for papers for the Black Hat Briefings USA (Las Vegas, 26–29 July 2004) will open 15 February 2004. For full details of all events, including information on how to submit a paper, see http://www.blackhat.com/.

The 13th Annual RSA Conference takes place in San Francisco from 23–27 February 2004. The aim of the RSA Conference is to bring together IT professionals, developers, policy makers, industry leaders and academics to share information and exchange ideas on technology trends and best practices in identity theft, hacking, cyber-terrorism, biometrics, network forensics, perimeter defence, secure web services, encryption and related topics. For more information see http://www.rsaconference.com/.

The NHTCU's Second e-Crime Congress will take place on the 24 and 25 February 2004 at the Victoria Park Plaza Hotel, London. Supported by the Home Office for the second year, the congress provides an opportunity for government, law enforcement and business to develop effective partnerships to address the threat of hi-tech crime. The e-Crime Congress aims to bring together 400 senior delegates from the public and private sectors. The theme of the congress is 'Designing Out Hi-Tech Crime', an examination of pre-emptive action. A series of interactive workshops will be held over the course of the two days, with the common goal of 'designing out' hi-tech crime. For more information including registration details, see http://www.e-crimecongress.org/.

The Open University will host a one-day anti-virus conference entitled 'Combating Vandalism in Cyberspace' on 4 March 2004 in Milton Keynes, UK. Registration costs £150 for corporate attendees and £100 for those from educational institutions. For programme and registration details see http://tscp.open.ac.uk/.

InfoSec World Conference and Expo 2004 takes place 22–24 March 2004 in Orlando, FL, USA. For details of the exhibition and a series of optional workshops see http://www.misti.com/.

Infosecurity Europe 2004 will be held from 27–29 April 2004 in the Grand Hall Olympia, London, UK. For all show details and registration enquiries see http://www.infosec.co.uk/.

The EICAR Conference 2004 will be held in Luxembourg City, from 1–4 May 2004. EICAR 2004 will feature only one stream, which will give in-depth coverage of issues including malware, critical infrastructure protection, legal and operational issues, and identity management and social issues. The call for papers remains open until 15 January 2004. More information, including guidelines for paper submission, is available from http://www.eicar.org/.

The 3rd Annual DallasCon Wireless Security Conference takes place 1–2 May 2004, in Dallas, TX, USA. The conference will feature two tracks: one track dedicated to the latest trends and threats in wireless security and a second track will focus on general information security. For details see http://www.dallascon.com/.

RSA Japan takes place 31 May to 1 June 2004 at the Akasaka Prince Hotel, Tokyo. For details see http://www.rsaconference.com/.

NetSec will take place 14–16 June 2004 in San Francisco, CA, USA. The conference program covers a broad array of topics, from the management issues of awareness, privacy and policy to more technical issues like wireless security, VPNs and Internet security. For full details see http://www.gocsi.com.

The 14th Virus Bulletin International Conference and Exhibition, VB2004, takes place 30 September – 1 October 2004 at the Fairmont Chicago, IL, USA. *Virus Bulletin* is currently seeking submissions from those wishing to present at the conference. For more information about the conference, including the full call for papers, and details of sponsorship and exhibition opportunities, see http://www.virusbtn.com/.

The 31st Annual Computer Security Conference and Expo will take place from 8–10 November 2004 at the Marriott Wardman Park in Washington, D.C., USA. More details will be available in due course from http://www.gocsi.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery: £195 (US$310)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England

Tel: +44 (0)1235 555139 Fax: +44 (0)1235 531889
Email: editorial@virusbtn.com www.virusbtn.com

vb Spam supplement

CONTENTS

NEWS & EVENTS

US AND UK LEGISLATION IN PLACE

While the 'CAN-SPAM Act' is expected to have been signed into US law by 1 January 2004, December 2003 saw the introduction of anti-spam legislation in the UK. Both sets of legislation have been criticised by members of the anti-spam community for making life easy for spammers – indeed, prolific spammer Alan Ralsky was reported to have said that the passage of the US bill through the House of Representatives 'made [his] day'. The concerns are that the US legislation fails to make spamming illegal, instead placing the onus on the recipient to opt out. Across the Atlantic, the UK's anti-spam legislation makes some forms of spamming illegal, but a (rather gaping) loophole allows spammers to continue targeting 'business' addresses unabated. Countries whose legislation has been met with a more positive response include Italy, where spamming is punishable by up to three years in jail, and Australia, where spammers may be fined up to $1.1 million a day. However, the effectiveness of any legislation in reducing the spam problem is likely to be countered while there remains such great disparity in anti-spam legislation across the world.

EVENTS

The NIST/CSD Spam Technology Workshop will be held on 17 February 2004 at NIST Gaithersburg Campus, USA. For full details see http://csrc.nist.gov/spam/.

101TechStrategies will hold an Anti-Spam Summit from 17–19 March 2004 in San Francisco, USA. For details see http://www.101techstrategies.com/.

OPINION

ECONOMIC AND LEGAL SOLUTIONS TO SPAM

Pete Sergeant

Following the 'CAN-SPAM Act' passed by the US Senate in late November 2003 (see *VB Spam Supplement*, November 2003 p.S1), there has been a flurry of commentaries on the use of legislation to curb spam. Many people have taken the perceived weaknesses in this legislation as a starting point for general rants on the (in)effectiveness of stopping spam through the power of law. Over and again, commentators have opined that radical changes to mail protocols and other technical solutions are the cure to the spam scourge. This article looks at ways to reduce spam that rely more heavily on social engineering.

First, let's take a look at the economics of spam. Infamous spammer Alan Ralsky said: 'I put people on the same playing field as Fortune 500 companies for a fraction of the cost.' As long as spam is effective at generating sales leads for a small cost, profit-seeking companies and retailers who are not worried about negative publicity will use it as a means to market their products, be they diet pills or diplomas – to do so makes sound economic sense.

Should spam cease to be a cost-effective way of advertising a product, economic theory suggests that the retailers will stop using this form of advertising. This situation will arise when demand for the product dries up, or when spammers are forced to raise their prices to a point at which the medium is too expensive.

It is not only companies selling products to whom this applies: when the cost of telling millions about your father, deposed despotic dictator of Djibouti, brings in fewer advances to aid the transfer of his funds to your American business associate than it cost you to send the emails, you may be inclined to stop informing the world of your plight and seek to pursue a more profitable scam.

With spam, there are three basic supply and demand situations: we have the spammers' demand for Internet connectivity and equipment, the retailers' demand for spammers' services, and finally, the end users' demand for the items being sold by retailers. In all cases, spam can be reduced by pushing up the costs for suppliers (thus reducing

their returns, and the price at which they are willing to sell), or by decreasing demand.

INFRASTRUCTURE

Let's start by looking at the cost to spammers of the spamming infrastructure – essentially server costs and bandwidth. Increasing the cost of servers for spammers would be impractical, but increasing the cost of the bandwidth is a viable option.

Some RBLs (Realtime Blackhole Lists), in particular the now-defunct SPEWS (Spam Prevention Early Warning System), did just this. SPEWS aimed for a large amount of collateral damage – if an ISP was tolerating spammers on its network (spammers chew a lot of bandwidth, generating a lot of revenue for the ISP), SPEWS would blacklist a large chunk of the ISP's network space with the result that legitimate customers found themselves unable to send email. When those customers discovered the problem was with their ISP, they often had no choice but to switch to another ISP so that their mail got through. As a result, demand for the ISP's services would drop and it would have to raise its prices, thus raising the costs for spammers, who would have to pass the costs on to their own clients – the use of spam thus becomes less profitable.

There are legal means to achieve the same effect: legislation that allows ISPs to sue other ISPs that persistently deposit large amounts of spam onto their networks suddenly causes an increase in the costs associated with an ISP allowing spammers on its network – thus forcing ISPs either to vet their customers more carefully and remove spammers, or hike up their costs. Again, these costs are passed on to those using spammers to sell their wares.

Finally, reputation goes a long way. If legitimate customers use their money wisely, and refuse to buy bandwidth from ISPs who have nefarious business practices with regards to spammers, or ISPs who don't vet their customers particularly carefully (information about 'spam-loving' ISPs is readily available online), those ISPs will be forced either to raise their prices or to force spammers off their network.

In summary: ISPs will deal with spammers because there is money to be made from them. If you make it more expensive for ISPs to sell bandwidth to spammers, or fail to vet their clients, it becomes harder and more expensive for spammers to operate – a cost that is passed on to their clients, and a step towards making spamming non cost-effective.

RETAILERS

Of course an easy target is the retailers and advertisers. In order to sell something via email, you have to provide some form of contact details. This makes tracing those who sell their wares via spam somewhat easier than it is to trace virus writers. In fact, the retailers are possibly the best place to aim legal action. If you can convince those selling their services that spamming will lead to heavy-handed 're-education' by the local law enforcement, you may significantly reduce their desire to sell by email. And, if you choke the demand, spamming becomes less profitable.

USER DEMAND

When users stop trying to increase their sperm count by 581%, or learn not to buy from people who send them unsolicited email, then demand falls – fewer units are sold, and so retailers gain less. Again, this decreases the cost-effectiveness of spamming.

As email-based scams receive more and more publicity, one hopes the number of people still gullible enough to buy from spam will drop. However, as more people go online, the target audience for spam will increase, perhaps negating this effect.

Hopefully the technical anti-spam solutions that are put in place by the big free email companies like *Hotmail* will go some way towards preventing users from seeing the spam in the first place. Beyond that, perhaps our only hope is education. The more anti-spam vendors and network administrators drill into their users that they should not send their money to some guy with an offer that sounds too good to be true, the better.

SPAMMERS

Finally, Lawrence Lessig, professor of law at Stanford University, suggests a simple law: force spammers (or advertisers who don't label their junk email) to pay $10,000 to the first recipient who finds them. This relieves law enforcement bodies of a lot of work and, of course, it raises someone's costs. If spammers regularly have to deal with legal fees and pay-outs, they will have to hike up their prices: retailers and advertisers will once again be forced to look elsewhere.

CONCLUSION

Until now, spamming has been looked upon by the majority as a technical problem that needs to be solved. By considering spam as an economic problem – in the same way that one considers something like pollution – we can aim to make spamming a commercially non-viable venture, and thus hope to end the cat-and-mouse game between those seeking technical solutions to spam and those looking to circumvent such measures.

SPOTLIGHT

AS THEY UNITE, SO MUST WE: COLLABORATING TO END THE SPAM EPIDEMIC

Paul Judge
CipherTrust, USA

For the past two months the VB Spam Supplement has included a summary of the ASRG mailing list. This month Paul Judge, the founder of the Anti-Spam Research Group, explains its aims and objectives.

The problem of spam – unwanted messages – has been around since the days of ARPANET and USENET. The problem has moved to the email world and it is also emerging in other mediums such as SMS and instant messaging.

Over the last couple of years, the size of the problem has grown almost exponentially. Just last year, spam accounted for only 10 per cent of inbound email traffic; today, spam accounts for more than 60 per cent of inbound email traffic. The scale and effect of the spam epidemic creates a costly problem and stands to impact the way that people use email and the Internet.

The severity of this problem has led many to focus on it over the years. In working on the spam problem, I realized that, while there were many intelligent people working on different facets of the problem in their own corners of the world, there was a lack of collaboration and an absence of a widely-known research agenda dedicated to solving the problem.

I also realized how the problem of spam held many similarities with other issues that technologists have faced. There was a need to introduce the spam problem to technologists from other areas.

ASRG

In March 2003, I chartered the Anti-Spam Research Group (ASRG) to provide a forum for contributions to anti-spam technology as well as expose the spam problem to researchers from other disciplines with a view to sparking collaboration and steering the anti-spam efforts around sound methodologies. These are the characteristics that have advanced technological development in other areas.

Three weeks after being chartered the ASRG held a physical meeting co-located with the 56th IETF (Internet Engineering Task Force) in March 2003. There were over 200 attendees at the first ASRG meeting and since then the group has grown to over 600 members, representing the various constituents of the email ecosystem including computer researchers, anti-spam companies, email server companies, email service providers, ISPs, anti-spam advocacy groups, end-users and administrators.

The ASRG focuses on technical solutions to the anti-spam problem. The group is guided by a research agenda that includes three phases: understanding the problem, proposing solutions, and evaluating solutions. While this may sound obvious, over the years, we, the anti-spam community, have been reacting to the problem and have oftentimes hurriedly proposed solutions without having spent as much effort first on understanding the problem that we were trying to solve or considering the limitations of our proposals.

Understanding the problem involves spam measurement, characterization, and analysis work. Evaluating solutions involves forming analysis criteria that explore the limitations, robustness to countermeasures, and deployment issues of proposals.

Based on this research agenda, the group has a work item list that includes about 20 research tasks and proposals, several of which have already been published as Internet Drafts. The ASRG is fulfilling its vision and is on track to make a number of meaningful contributions to anti-spam technology.

It has become evident that while anti-spam technology forms the core of local spam solutions, technology must be coupled with other efforts including legislative and user education/awareness efforts in order to solve the global spam problem. While the ASRG focuses on technological solutions, other anti-spam organizations and initiatives such as SpamCon (http://www.spamcon.org/) and the Coalition Against Unsolicited Commercial Email (CAUCE, http://www.cauce.org/) focus on education/awareness and advocacy efforts, respectively.

COLLABORATION

I applaud *Virus Bulletin*'s efforts to bring information about the spam problem to its audience in the *VB Spam Supplement*. This type of inter-disciplinary awareness and involvement is quite necessary.

There are a number of parallels between the spam and virus problems. This has become increasingly evident over the last year with the spread of worms such as SoBig. There are many lessons that the anti-spam and anti-virus communities can learn from each other. There is a pressing need for collaboration – not only in responding to these threats but also in developing new solutions. As our adversaries unite, the line between these problems dissolves and we find ourselves as one community fighting a single evolving threat.

SUMMARY

ASRG SUMMARY: DECEMBER 2003

Pete Sergeant

At the tail end of November, Sandeep Krishnamurthy made an appeal to ASRG members to help with some academic research by completing a spam survey. The survey prompted a number of comments with the result that the discussion moved on to problems concerned with reminding users that they have actually opted in for certain mailings. Alan DeKok suggested some form of token exchange system as a possible solution to the problem, but pointed out that such a system would need to be believed by and accountable to both parties. Markus Stumpf said that this exists already in the form of subscription confirmations, but people delete these, so he questioned why any other system would be more effective.

Jon Kyme noted that the 'CAN-SPAM Act' requires the labelling of commercial emails, and that there is no IETF standard for this. A number of members stepped up to offer their help. 'Chris' felt that the labelling should be required to give some indication of what is being sold – he indicated that, while he might be happy to receive spam concerning carpentry tools, he would not want emails advertising women's personal products. Eric Brunner-Williams suggested people examine PICS (Platform for Internet Content Selection, http://www.w3.org/PICS/) and P3P (Platform for Privacy Preferences, http://www.w3.org/P3P) for prior work on the subject.

Eric S. Raymond (ESR) then posted his Internet-Draft on standards for labelling of commercial and bulk email. His draft insists on the letters 'ADV' in the subject line of bulk commercial emails, and suggests the use of the words 'porn' and 'advertising' in addition to 'bulk' in the 'Keywords' header.

ESR's draft generated some 40-odd replies. People questioned whether 'ADV' was sufficiently cross-encoding, and clarification was requested concerning whether 'ADV' was searched for before or after decoding. Phillip Hallam-Baker pointed out that there was already some consensus in the use of 'RE' and other labels to denote replies and so on.

ESR's update clarifies: 'Internationalization was considered. With a total inventory of four tokens, the benefits seem small and the added complexity of a multilingual vocabulary rather pointless.'

Walter Dnes brought up the subject of using RSS as a replacement for mailing lists, so that emails are sent to a list server, from where it is available by RSS for subscribers. People drew parallels to *PointCast*, but Walter said he believes *PointCast* failed because it was proprietary and tried to ram ads down the throats of people who didn't want them.

Yakov Shafranovich questioned whether big marketers would be likely to switch, but again Walter thought not: 'The good part of pull, as far as I'm concerned, is that control rests with the end user, not the marketer. The bad part of pull, as far as the marketer is concerned, is that control rests with the end user, not the marketer.'

Yakov posted a message to the main ASRG list that had been originated by Philip Miller on the Best Current Practices list (which is shortly to be closed down, with the archive added to the main ASRG site and discussions on the BCP area moved to the main ASRG list).

Philip was aiming to compile a list of current classes of people who send mails (think 'actors' in use-case diagrams), so that draft BCP recommendations can be created for each of them. To quote: 'Once we have a relatively comprehensive list, we should assemble all of the "common sense" best practices that have been somewhat implicit in discussions on this list, but may not be eminently apparent to any random person. We can collect them for draft publication, but we should probably evaluate them first in light of their effect.'

Yakov picked up on Hector Santos's posting of an article discussing *Yahoo*'s PKI initiative to stop spam. Mark Baugher echoed the sentiments of an article in *Virus Bulletin* about the problems caused by viruses that steal private keys (see *VB*, September 2003, p.12) – although Alan DeKok pointed out that this allows you to see whose machine is infected.

Yakov also suggested the creation of a new subgroup to discuss push vs pull technology such as RSS-based newsgroups. Yakov's mail also summed up the month's discussion on pull technology very nicely:

'There are two arguments that were put forth in the list discussion. One contingent was arguing that providing "pull" channels for marketers, can help mitigate spam if most legit marketers switch over to "pull" leaving only spammers in the email realm. However, it seems to us that until "pull" standards are improved and there is better integration with email, that will not happen. The second contingent was arguing for overhauling the email infrastructure to do pull instead of push (this would include IM2000 proposal for example).'

Yakov went on to describe how these require changes to pull technology or email standards, and said that this does not really fall under the scope of the ASRG lists, asking that the discussion be taken elsewhere.